Compressive Sensing

Dr.-Ing. Ali Bereyhi

Friedrich-Alexander University

ali.bereyhi@fau.de

Summer 2022

Where Are We?

We ended up with following results

- *The optimal approach for sparse recovery is*

  ℓ_0-norm minimization

- *For unique sparse recovery via the optimal approach at least*

$$s \leq M^{\star} \leq 2s$$

  samples should be collected

- *ℓ_0-norm minimization is an NP-hard problem*

  It cannot be implemented in practice

Where Are We?

Now, we want to check out

some sparse recovery algorithms

which can be used in practice

Algorithms for Sparse Recovery

Sparse Recovery: *Suboptimal Approaches*

There is a general simple rule

Unlike optimal approach, suboptimal approaches are not unique

So how shall we choose a suboptimal algorithm?

There is always a trade-off in between

- *An algorithm might require less number of measurements*
- *It then probably is computationally more complex*

So, it really depends on our budget!

Sparse Recovery: *Suboptimal Approaches*

What are the suboptimal approaches?

In general, anyone can come up with an approach!

However, there are popular algorithms with good trade-off

- *ℓ_1-Norm Minimization Algorithms*
- *Greedy Algorithms*
- *Iterative Thresholding Algorithms*
- *AMP-based Algorithms*

We learn the last one in the second part of the lecture

ℓ_1-Norm Minimization Algorithms

Approximating Optimal Solution via ℓ_1-Norm

What do we do in this approach?

In a nutshell,

We replace the ℓ_0-norm with the ℓ_1-norm

But why should it work fine?

We see it now!

Back to Our First Example

Remember the example in the introductory part

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

We showed that the set of all solutions is

$$\boldsymbol{x}(u) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 - 2u \\ 1 + u \\ u \\ 1 - u \end{bmatrix}$$

Back to Our First Example

Remember the example in the introductory part

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

The sparsest solution is the 2-sparse vector $\boldsymbol{x}\,(u = 1)$

$$\boldsymbol{x}\,(u = 1) = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Back to Our First Example

Remember the example in the introductory part

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

This means that $\boldsymbol{x}\left(u=1\right)$

$$\boldsymbol{x}\left(u=1\right) = \operatorname{argmin} \|\boldsymbol{z}\|_0 \;\; \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Back to Our First Example

Remember the example in the introductory part

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

Or equivalently,

$$\boldsymbol{x}(u = 1) = \operatorname*{argmin}_{\boldsymbol{x}(u) \text{ for } u \in \mathbb{R}} \|\boldsymbol{x}(u)\|_0$$

Let's see how $\|\boldsymbol{x}(u)\|_0$ *looks against* u

Back to Our First Example

At $u = -1$, $u = 0$ *and* $u = 1$, $\boldsymbol{x}(u)$ *is*

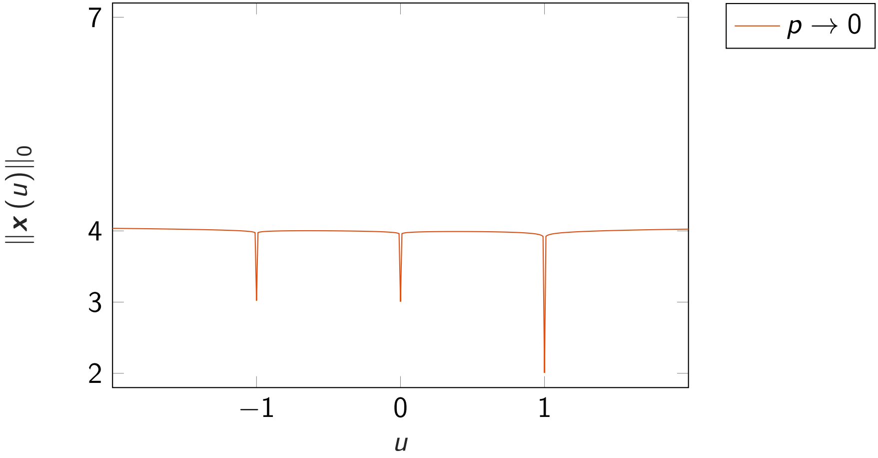
$$\boldsymbol{x}(u = -1) = \begin{bmatrix} 4 \\ 0 \\ -1 \\ 2 \end{bmatrix}, \ \boldsymbol{x}(u = 0) = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \ \boldsymbol{x}(u = 1) = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

which means

$$\|\boldsymbol{x}(u = -1)\|_0 = 3, \ \|\boldsymbol{x}(u = 0)\|_0 = 3, \ \|\boldsymbol{x}(u = 1)\|_0 = 2$$

For any other choice of u, $\|\boldsymbol{x}(u)\|_0 = 4$

Back to Our First Example



Back to Our First Example

We know that the key issue is with

$$\boxed{\ell_0\text{-norm function}}$$

So, what if we relax it with an ℓ_p-norm?

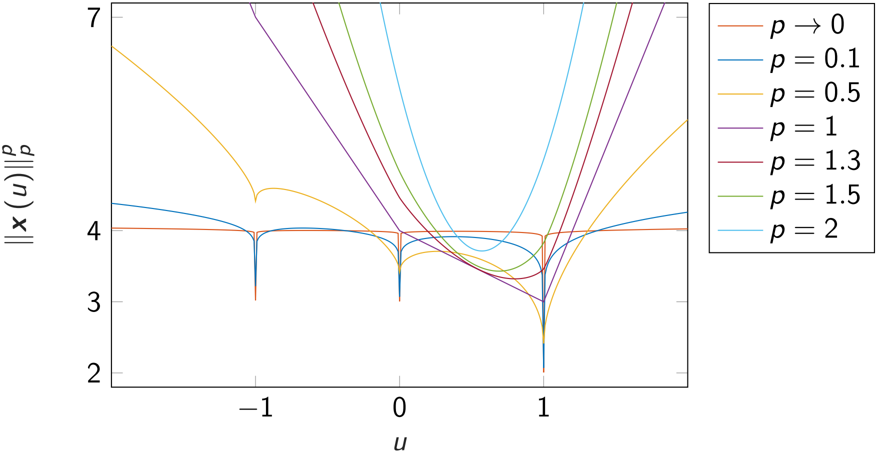
Let's do an experiment: We try to see how does

$$\|\boldsymbol{x}(u)\|_p^p$$

look like for different choices of p

Relaxation means we replace the problematic part with an *approximation*

Back to Our First Example



Approximating Optimal Solution via ℓ_1-Norm

From the figure, one can see that

$$\boxed{\textit{For } 0 \leq p \leq 1 \textit{ the global minimum occur at the same } u}$$

But . . .

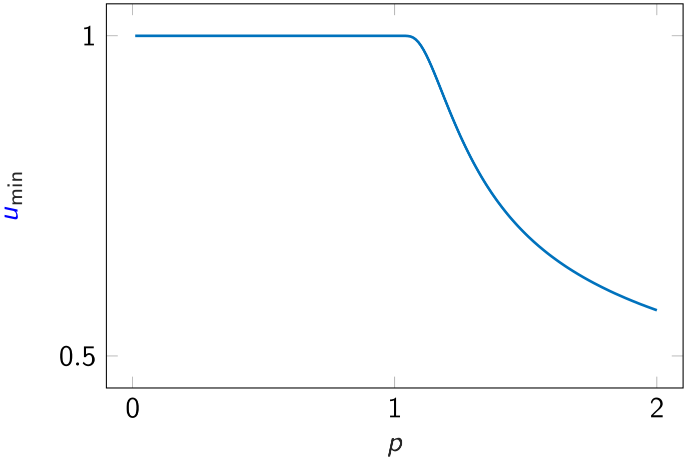
$$\boxed{\textit{As } p > 1 \textit{ the global minimizer starts to move}}$$

To see this clearly, let us plot

$$u_{\min} = \underset{u}{\operatorname{argmin}} \|\boldsymbol{x}(u)\|_p^p$$

against p

Approximating Optimal Solution via ℓ_1-Norm



Approximating Optimal Solution via ℓ_1-Norm

This is in fact a general property

> *ℓ_p-norm minimization recovers a sparse vector for $0 \leq p \leq 1$*

What would be then a good choice for relaxation?

- *For $0 \leq p \leq 1$, we can recover a sparse vector*
- *For $p \geq 1$, we deal with convex optimization*

Thus, the best choice for relaxation is

ℓ_1-norm

ℓ_1-Norm Minimization Algorithms

Section 1: Basis Pursuit Algorithm

Basis Pursuit

Basis Pursuit

Basis pursuit recovers the signal from samples in $\boldsymbol{y}$ *by*

$$\min \|\boldsymbol{z}\|_1 \ \ \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Although we have seen it intuitively, we still want to make sure

Basis pursuit recovers a sparse signal

and know what is the sparsity level

Basis Pursuit: *How Sparse is Recovery?*

An Initial Claim

For $\mathbf{A} \in \mathbb{R}^{M \times N}$ *basis pursuit recovers* $\boldsymbol{x}^\star$

$$\boldsymbol{x}^\star = \operatorname{argmin} \|\boldsymbol{z}\|_1 \ \textit{ subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Then, $\boldsymbol{x}^\star$ *has at most* M *non-zero elements*

To show that this claim is correct consider this reminder

The kernel of a fat matrix has members other than $\mathbf{0}$

$$\ker \ \textit{fat}\ \mathbf{A} \neq \{\mathbf{0}\}$$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

Clearly, $\boldsymbol{x}^\star$ *satisfies the linear system of equations*

$$\mathbf{A}\boldsymbol{x}^\star = \boldsymbol{y} \rightsquigarrow \mathbf{A}_{\mathcal{S}}\boldsymbol{x}^\star_{\mathcal{S}} = \boldsymbol{y}$$

The submatrix $\mathbf{A}_{\mathcal{S}}$ *is of size* $M \times s$ *which is fat*

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\mathbf{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\mathbf{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

Since $\mathbf{A}_\mathcal{S}$ *is fat, we can find non-zero* $\mathbf{v} \in \mathbb{R}^s$ *such that*

$$\mathbf{v} \in \ker \mathbf{A}_\mathcal{S} \rightsquigarrow \mathbf{A}_\mathcal{S}\mathbf{v} = 0$$

We now construct a new vector $\mathbf{x}^\sharp = \mathbf{x}^\star + t\tilde{\mathbf{v}}$

$$\boxed{\tilde{\mathbf{v}} \in \mathbb{R}^N \text{ is } s\text{-sparse with support } \mathcal{S} \text{ and } \tilde{\mathbf{v}}_\mathcal{S} = \mathbf{v}}$$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

Well, clearly we have

$$\begin{aligned} \mathbf{A}\boldsymbol{x}^\sharp &= \mathbf{A}\left(\boldsymbol{x}^\star + t\tilde{\boldsymbol{v}}\right) \\ &= \mathbf{A}_{\mathcal{S}}\left(\boldsymbol{x}^\star_{\mathcal{S}} + t\tilde{\boldsymbol{v}}_{\mathcal{S}}\right) = \boldsymbol{y} + t\,\mathbf{0} = \boldsymbol{y} \end{aligned}$$

Now, what about its $\ell_1 -$ *norm?*

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

We now determine the ℓ_1*-norm of* $\boldsymbol{x}^\sharp$

$$\begin{aligned}\|\boldsymbol{x}^\sharp\|_1 &= \|\boldsymbol{x}^\star + t\tilde{\boldsymbol{v}}\|_1 \\ &= \sum_{n=1}^{N} |x_n^\star + t\tilde{v}_n| = \sum_{n\in\mathcal{S}} |x_n^\star + t\tilde{v}_n|\end{aligned}$$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

We first note that

$$\text{If} \quad |t| < \frac{|x_n^\star|}{|\tilde{v}_n|} \rightsquigarrow |t\tilde{v}_n| < |x_n^\star|$$

In this case, the sign of $x_n^\star + t\tilde{v}_n$ *is same as the sign of* $x_n^\star$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

Now, we focus on an interval of t in which

$$|t| < \min_{n \in \mathcal{S}} \frac{|x_n^\star|}{|\tilde{v}_n|} := \theta$$

With $|t| < \theta$, *we have* $|x_n^\star + t\tilde{v}_n| = \operatorname{Sgn}(x_n^\star)(x_n^\star + t\tilde{v}_n)$ *for* $n \in \mathcal{S}$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

The ℓ_1*-norm of* $\boldsymbol{x}^\sharp$ *is hence written for* $|t| < \theta$ *as*

$$\begin{aligned}\|\boldsymbol{x}^\sharp\|_1 &= \sum_{n \in \mathcal{S}} \mathrm{Sgn}\left(x_n^\star + t\tilde{v}_n\right)\left(x_n^\star + t\tilde{v}_n\right) \\ &= \sum_{n \in \mathcal{S}} \mathrm{Sgn}\left(x_n^\star\right)\left(x_n^\star + t\tilde{v}_n\right)\end{aligned}$$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ with $\mathcal{S}$, where $|\mathcal{S}| = s$

The ℓ_1-norm of $\boldsymbol{x}^\sharp$ is hence written for $|t| < \theta$ as

$$\begin{aligned}\|\boldsymbol{x}^\sharp\|_1 &= \sum_{n\in\mathcal{S}} \mathrm{Sgn}\left(x_n^\star\right)\left(x_n^\star + t\tilde{v}_n\right) \\ &= \sum_{n\in\mathcal{S}} \mathrm{Sgn}\left(x_n^\star\right) x_n^\star + t \sum_{n\in\mathcal{S}} \mathrm{Sgn}\left(x_n^\star\right) \tilde{v}_n\end{aligned}$$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

We note that

$$\sum_{n \in \mathcal{S}} \mathrm{Sgn}\left(x_n^\star\right) x_n^\star = \sum_{n \in \mathcal{S}} \left|x_n^\star\right|$$
$$= \left\|\boldsymbol{x}^\star\right\|_1$$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ with $\mathcal{S}$, where $|\mathcal{S}| = s$

So, we can write

$$\|\boldsymbol{x}^\sharp\|_1 = \|\boldsymbol{x}^\star\|_1 + t \sum_{n \in \mathcal{S}} \operatorname{Sgn}\left(x_n^\star\right) \tilde{v}_n$$

for all $|t| \leq \theta$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^\star \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^\star$ *with* $\mathcal{S}$, *where* $|\mathcal{S}| = s$

Interval $|t| \leq \theta$ *always include a real number* t_1 *such that*

$$t_1 \sum_{n \in \mathcal{S}} \operatorname{Sgn}\left(x_n^\star\right) \tilde{v}_n < 0$$

This means that we can always find $\boldsymbol{x}^\sharp$, *such that* $\|\boldsymbol{x}^\sharp\|_1 < \|\boldsymbol{x}^\star\|_1$

Basis Pursuit: *How Sparse is Recovery?*

Let's start with the opposite assumption

$$\boldsymbol{x}^{\star} \text{ has } s > M \text{ non-zero elements}$$

We show the support of $\boldsymbol{x}^{\star}$ with $\mathcal{S}$, where $|\mathcal{S}| = s$

This is now a contradiction with the fact that

$$\boldsymbol{x}^{\star} \text{ is the global minimizer of } \ell_1\text{-norm on the feasible set}$$

Our initial claim is thus proven by contradiction

Basis Pursuit: *How Sparse is Recovery?*

To avoid the contradiction what do we need?

- *First of all* $\boldsymbol{x}^\star$ *be s-sparse with* $s \leq M$
- *Secondly, for* $\mathbf{A}$ *we need to have*

$$\ker \mathbf{A}_{\mathcal{S}} = \{\mathbf{0}\}$$

for the recovery support $\mathcal{S}$ *with* $|\mathcal{S}| \leq s$

Sparsity of Basis Pursuit Recovery

Basis pursuit always recovers a s-sparse signal with $s \leq M$ *and the columns of* $\mathbf{A}_{\mathcal{S}}$ *are linearly independent*

Summary

Up to this point, we have learned that

- *Basis pursuit $\equiv$ ℓ_1-norm minimization gives*

  Best relaxation of the optimal sparse recovery

- *We can make sure that basis pursuit recovers*

  A signal whose sparsity is at most as much as # of samples

Now the question is that

- *How can we implement the basis pursuit algorithm?*

$\boxed{\textit{We study it in the next section}}$

Implementing Basis Pursuit Algorithm

Getting Back to Basis Pursuit

Basis Pursuit

Basis pursuit recovers the signal from samples in $\boldsymbol{y}$ by

$$\min \|\boldsymbol{z}\|_1 \ \ \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

One question which comes to mind

$$\boxed{\textit{Is basis pursuit, the only relaxation via } \ell_1\textit{-norm?}}$$

We see shortly that the answer is No!

Implementing Basis Pursuit Algorithm

Section 1: Alternative Forms of ℓ_1-Norm Minimization

Basis Pursuit with Quadratic Constraint

Basis Pursuit

$$\min \|\boldsymbol{z}\|_1 \ \ \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

The linear constraint in basis pursuit can be translated as

$$\mathbf{A}\boldsymbol{z} = \boldsymbol{y} \leftrightsquigarrow \mathbf{A}\boldsymbol{z} - \boldsymbol{y} = \mathbf{0} \leftrightsquigarrow \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 = 0$$

Basis Pursuit with Quadratic Constraint

Basis Pursuit

$$\min \|\boldsymbol{z}\|_1 \quad \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Now, we consider the following quadratic constraint

$$\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \eta$$

What happens if $\eta \to 0$*?*

$$\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \eta \leftrightsquigarrow \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 = 0 \leftrightsquigarrow \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Basis Pursuit with Quadratic Constraint

Basis Pursuit

$$\min \|\boldsymbol{z}\|_1 \ \ \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

So, we can alternatively solve the following problem and let $\eta \to 0$

Basis Pursuit with Quadratic Constraint

$$\min \|\boldsymbol{z}\|_1 \ \ \textit{subject to } \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \eta$$

Basis Pursuit with Quadratic Constraint

Basis Pursuit with Quadratic Constraint

$$\min \|\boldsymbol{z}\|_1 \;\; \textit{subject to } \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \eta$$

Attention!

You might think

$$\eta \to 0 \textit{ is the only meaningful choice}$$

But, as we see later

Other choices of η are also useful in noisy settings

Basis Pursuit Denoising

Basis Pursuit

$$\min \|\boldsymbol{z}\|_1 \;\; \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

Let us construct a new objective function $F_\lambda(\boldsymbol{z})$ as below

$$F_\lambda(\boldsymbol{z}) = \frac{1}{2\lambda}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \|\boldsymbol{z}\|_1$$

for some real $\lambda \geq 0$

Basis Pursuit Denoising

Basis Pursuit

$$\min \|\boldsymbol{z}\|_1 \;\; \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

What happens for this objective, if $\lambda \to 0$?

$$F_{\lambda}(\boldsymbol{z}) = \frac{1}{2\lambda}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \|\boldsymbol{z}\|_1 = \begin{cases} \|\boldsymbol{z}\|_1 & \textit{if } \mathbf{A}\boldsymbol{z} = \boldsymbol{y} \\ +\infty & \textit{if } \mathbf{A}\boldsymbol{z} \neq \boldsymbol{y} \end{cases}$$

Basis Pursuit Denoising

Basis Pursuit

$$\min \|\mathbf{z}\|_1 \ \ \textit{subject to } \mathbf{A}\mathbf{z} = \mathbf{y}$$

So, if the solution of basis pursuit has a finite ℓ_1-norm, we have

$$\left\{ \lim_{\lambda \to 0} \min F_\lambda(\mathbf{z}) \right\} = \min \left\{ \|\mathbf{z}\|_1 \ \ \textit{subject to } \mathbf{A}\mathbf{z} = \mathbf{y} \right\} \cup \{+\infty\}$$
$$= \left\{ \min \|\mathbf{z}\|_1 \ \ \textit{subject to } \mathbf{A}\mathbf{z} = \mathbf{y} \right\}$$

Basis Pursuit Denoising

Basis Pursuit

$$\min \|\boldsymbol{z}\|_1 \quad \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

We could in fact recover the basis pursuit solution by

Basis Pursuit Denoising

$$\min \frac{1}{2}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

Basis Pursuit Denoising

From Denoising to Pure Basis Pursuit

Let $\boldsymbol{x}^\star$ and $\boldsymbol{x}(\lambda)$ be defined as

$$\boldsymbol{x}^\star = \operatorname{argmin} \|\boldsymbol{z}\|_1 \ \ \textit{subject to } \mathbf{A}\boldsymbol{z} = \boldsymbol{y}$$

$$\boldsymbol{x}(\lambda) = \operatorname{argmin} \frac{1}{2}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

Then, if $\boldsymbol{x}^\star$ is the unique solution, we have

$$\lim_{\lambda \to 0} \boldsymbol{x}(\lambda) = \boldsymbol{x}^\star$$

Basis Pursuit Denoising

How can we show this equivalency?

Before, we start with the proof, remember that

$$\boldsymbol{x}^\star \text{ is the solution of basis pursuit}$$

So, for any solution $\boldsymbol{x}$ *of the underdetermined equation, i.e.,*

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$$

We have

$$\|\boldsymbol{x}\|_1 \geq \|\boldsymbol{x}^\star\|_1$$

Basis Pursuit Denoising

How can we show this equivalency?

First of all, since $\boldsymbol{x}^\star$ *is the basis pursuit solution, we have*

$$\mathbf{A}\boldsymbol{x}^\star = \boldsymbol{y}$$

This means that

$$\frac{1}{2}\underbrace{\|\mathbf{A}\boldsymbol{x}^\star - \boldsymbol{y}\|_2^2}_{0} + \lambda\|\boldsymbol{x}^\star\|_1 = \lambda\|\boldsymbol{x}^\star\|_1$$

Basis Pursuit Denoising

How can we show this equivalency?

If we define

$$G_{\lambda}(\boldsymbol{z}) = \frac{1}{2}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

Our conclusion simply says that

$$G_{\lambda}(\boldsymbol{x}^{\star}) = \lambda\|\boldsymbol{x}^{\star}\|_1$$

Basis Pursuit Denoising

How can we show this equivalency?

Secondly, we know that

$$\boldsymbol{x}(\lambda) \text{ is the minimizer of } G_\lambda(\boldsymbol{z})$$

This means that for any $\boldsymbol{z}$

$$G_\lambda(\boldsymbol{x}(\lambda)) \leq G_\lambda(\boldsymbol{z})$$

This is also true for $\boldsymbol{z} = \boldsymbol{x}^\star$, *i.e.,* $G_\lambda(\boldsymbol{x}(\lambda)) \leq G_\lambda(\boldsymbol{x}^\star)$ *or*

$$\frac{1}{2}\|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}(\lambda)\|_1 \leq \lambda\|\boldsymbol{x}^\star\|_1$$

Basis Pursuit Denoising

How can we show this equivalency?

Noting that

$$\|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2,\ \lambda\|\boldsymbol{x}(\lambda)\|_1 \text{ and } \lambda\|\boldsymbol{x}^\star\|_1 \text{ are non-negative}$$

We can conclude out of the last inequality that

$$\lambda\|\boldsymbol{x}(\lambda)\|_1 \leq \lambda\|\boldsymbol{x}^\star\|_1 \rightsquigarrow \|\boldsymbol{x}(\lambda)\|_1 \leq \|\boldsymbol{x}^\star\|_1$$

$$\frac{1}{2}\|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2 \leq \lambda\|\boldsymbol{x}^\star\|_1$$

Basis Pursuit Denoising

How can we show this equivalency?

We know at the moment that

$$\|\boldsymbol{x}(\lambda)\|_1 \leq \|\boldsymbol{x}^\star\|_1$$

which is also true if $\lambda \to 0$. *Now, remember that*

$$\frac{1}{2}\|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2 \leq \lambda\|\boldsymbol{x}^\star\|_1$$

If we send $\lambda \to 0$, *we conclude that*

$$\lim_{\lambda \to 0} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2 \leq 0 \rightsquigarrow \lim_{\lambda \to 0} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2 = 0$$

Basis Pursuit Denoising

How can we show this equivalency?

The identity

$$\lim_{\lambda \to 0} \frac{1}{2} \|\mathbf{A}\boldsymbol{x}(\lambda) - \boldsymbol{y}\|_2^2 = 0$$

means that if $\boldsymbol{x}(\lambda)$ *converges as* $\lambda \to 0$, *we have*

$$\mathbf{A} \lim_{\lambda \to 0} \boldsymbol{x}(\lambda) = \boldsymbol{y}$$

In other words,

$\lim_{\lambda \to 0} \boldsymbol{x}(\lambda)$ *is a solution of the underdetermined equation*

Basis Pursuit Denoising

How can we show this equivalency?

Well, we could say that

$$\|\lim_{\lambda \to 0} \boldsymbol{x}(\lambda)\|_1 \geq \|\boldsymbol{x}^\star\|_1$$

Put it beside the fact that

$$\|\lim_{\lambda \to 0} \boldsymbol{x}(\lambda)\|_1 \leq \|\boldsymbol{x}^\star\|_1$$

We need to have

$$\|\lim_{\lambda \to 0} \boldsymbol{x}(\lambda)\|_1 = \|\boldsymbol{x}^\star\|_1$$

Basis Pursuit Denoising

How can we show this equivalency?

The uniqueness of $\boldsymbol{x}^\star$ finally concludes that

$$\lim_{\lambda \to 0} \boldsymbol{x}(\lambda) = \boldsymbol{x}^\star$$

Attention!

- *Again, you can think of $\lambda \neq 0$ in noisy settings*
- *This alternative form is a key form, we use for implementation*

LASSO

Basis Pursuit with Quadratic Constraint

$$\min \|\mathbf{z}\|_1 \quad \textit{subject to } \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 \leq \eta$$

Assume that we have the unique solution $\mathbf{x}^\star$

Assume that $\mathcal{Z}$ *is the following set*

$$\mathcal{Z} = \left\{ \textit{all } \mathbf{z} \in \mathbb{R}^N : \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 \leq \eta \right\}$$

LASSO

Basis Pursuit with Quadratic Constraint

$$\min \|\boldsymbol{z}\|_1 \quad \textit{subject to } \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \eta$$

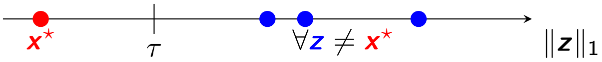
Assume that we have the unique solution $\boldsymbol{x}^\star$

We thus have

$$\boldsymbol{x}^\star = \underset{\boldsymbol{z} \in \mathcal{Z}}{\operatorname{argmin}} \|\boldsymbol{z}\|_1$$

LASSO

What happens if we plot the ℓ_1-norm of all $\mathbf{z} \in \mathcal{Z}$?



- *$\mathbf{x}^\star$ is the unique solution*
- *$\mathbf{z}$ are all other points in $\mathcal{Z}$*

We could now think of one τ, for which

$$\|\mathbf{x}^\star\|_1 \leq \tau$$
$$\|\mathbf{z}\|_1 > \tau$$

LASSO

We could hence say,

$$\{\boldsymbol{x}^\star\} = \mathcal{Z} \cap \{\text{all } \boldsymbol{z} : \|\boldsymbol{z}\|_1 \leq \tau\}$$

What about the other points $\boldsymbol{z} \neq \boldsymbol{x}^\star$ *in* $\{\text{all } \boldsymbol{z} : \|\boldsymbol{z}\|_1 \leq \tau\}$*?*

Well, since $\boldsymbol{x}^\star$ *is the unique solution, we have*

$$\forall \{\text{all } \boldsymbol{z} : \|\boldsymbol{z}\|_1 \leq \tau\} \ni \boldsymbol{z} \neq \boldsymbol{x}^\star : \boldsymbol{z} \notin \mathcal{Z}$$

This means

$$\forall \{\text{all } \boldsymbol{z} : \|\boldsymbol{z}\|_1 \leq \tau\} \ni \boldsymbol{z} \neq \boldsymbol{x}^\star : \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 > \eta$$

LASSO

So, we could break $\{\text{all } \mathbf{z} : \|\mathbf{z}\|_1 \leq \tau\}$ *into two cases*

$$\{\text{all } \mathbf{z} : \|\mathbf{z}\|_1 \leq \tau\} : \begin{cases} \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 > \eta & \mathbf{z} \neq \mathbf{x}^\star \\ \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 \leq \eta & \mathbf{z} = \mathbf{x}^\star \end{cases}$$

This means that

$$\mathbf{x}^\star = \operatorname{argmin} \|\mathbf{A}\mathbf{z} - \mathbf{y}\|_2^2 \ \textit{ subject to } \|\mathbf{z}\|_1 \leq \tau$$

This was discovered by Robert Tibshirani in 1996 under the name

Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO

LASSO

$$\min \|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \;\; \textit{subject to} \;\; \|\boldsymbol{z}\|_1 \leq \tau$$

Attention!

- *The value of τ is determined in terms of η*
- *LASSO is the oldest ℓ_1-norm minimization technique*

Implementing Basis Pursuit Algorithm

Section 2: The Homotopy Method

Homotopy in Nutshell

Remember that we can perform basis pursuit by denoising as

$$\boldsymbol{x}(\lambda) = \operatorname{argmin} \frac{1}{2}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

and then taking the limit $\lambda \to 0$

The homotopy method uses this property

- *It first finds* $\lambda^{(0)}$ *such that* $\boldsymbol{x}\left(\lambda^{(0)}\right) = \mathbf{0}$
- *It then step by step updates* $\boldsymbol{x}\left(\lambda^{(t)}\right)$ *till it gets to* $\lambda^{(t)} = 0$

The Homotopy Method: *Derivation*

Noting that the function

$$G_{\lambda}(\boldsymbol{z}) = \frac{1}{2}\|\mathbf{A}\boldsymbol{z} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{z}\|_1$$

is convex, we can find its minimizer by checking the extreme points

Let $\boldsymbol{x}(\lambda)$ *be an extreme point; then, we find for* $n \in \{1, \dots, N\}$

$$\left[\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{x}(\lambda) - \mathbf{A}^\mathsf{T}\boldsymbol{y}\right]_n + \lambda \mathrm{Sgn}(x_n(\lambda)) = 0 \quad \forall n : x_n(\lambda) \neq 0$$

$$\left|\left[\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{x}(\lambda) - \mathbf{A}^\mathsf{T}\boldsymbol{y}\right]_n\right| \leq \lambda \quad \forall n : x_n(\lambda) = 0$$

$[\boldsymbol{a}]_n$ *is the n-th entry of* $\boldsymbol{a}$ *and* $\mathrm{Sgn}(x)$ *is the sign of* x

The Homotopy Method: *Derivation*

To have $\mathbf{x}\left(\lambda^{(0)}\right) = \mathbf{0}$, *we need to choose* $\lambda^{(0)}$, *such that*

$$\left| \left[\mathbf{A}^{\mathsf{T}} \mathbf{A} \boldsymbol{x} \left(\lambda^{(0)} \right) - \mathbf{A}^{\mathsf{T}} \boldsymbol{y} \right]_n \right| \leq \lambda^{(0)} \rightsquigarrow \left| \left[\mathbf{A}^{\mathsf{T}} \boldsymbol{y} \right]_n \right| \leq \lambda^{(0)}$$

for all $n \in \{1, \ldots, N\}$. *One good choice is*

$$\lambda^{(0)} = \left| \left[\mathbf{A}^{\mathsf{T}} \boldsymbol{y} \right]_{n_1} \right|$$

where n_1 *is*

$$n_1 = \underset{n}{\operatorname{argmax}} \left| \left[\mathbf{A}^{\mathsf{T}} \boldsymbol{y} \right]_n \right|$$

The Homotopy Method: *Derivation*

Let us now initiate our algorithm

We start at point $\boldsymbol{x}^{(0)} = \mathbf{0}$, *and set*

$$\lambda^{(0)} = \max_n \left| \left[\mathbf{A}^\mathsf{T} \boldsymbol{y} \right]_n \right|$$

For compactness, we further define the residual in iteration t

$$\boldsymbol{r}^{(t)} = \mathbf{A}^\mathsf{T} \left(\mathbf{A} \boldsymbol{x}^{(t)} - \boldsymbol{y} \right)$$

For instance at $t = 0$, *we have* $\boldsymbol{r}^{(0)} = -\mathbf{A}^\mathsf{T} \boldsymbol{y}$

The Homotopy Method: *Derivation*

So, we could say

> *We start at point* $\boldsymbol{x}^{(0)} = \mathbf{0}$, *and set*
>
> $$\lambda^{(0)} = \max_n \left| r_n^{(0)} \right|$$
>
> *and denote the index of largest residual element with* n_1

where we consider

$$\boldsymbol{r}^{(t)} = \begin{bmatrix} r_1^{(t)} \\ \vdots \\ r_N^{(t)} \end{bmatrix}$$

The Homotopy Method: *Derivation*

Remember the extreme point equations

$$\boxed{\begin{array}{ll} \left[\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{x}\left(\lambda\right) - \mathbf{A}^\mathsf{T}\boldsymbol{y}\right]_n + \lambda \mathrm{Sgn}\left(x_n\left(\lambda\right)\right) = 0 & \forall n : x_n\left(\lambda\right) \neq 0 \\ \left|\left[\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{x}\left(\lambda\right) - \mathbf{A}^\mathsf{T}\boldsymbol{y}\right]_n\right| \leq \lambda & \forall n : x_n\left(\lambda\right) = 0 \end{array}}$$

Note that at iteration t with $\lambda^{(t)}$ and $\boldsymbol{x}^{(t)} = \boldsymbol{x}\left(\lambda^{(t)}\right)$, we have

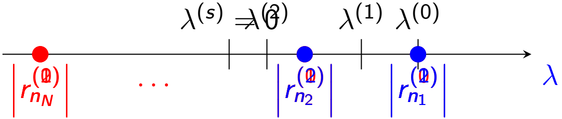
$$\boldsymbol{r}^{(t)} = \mathbf{A}^\mathsf{T}\left(\mathbf{A}\boldsymbol{x}^{(t)} - \boldsymbol{y}\right)$$

The Homotopy Method: *Derivation*

So, in iteration t the equations read

$$
\boxed{
\begin{array}{ll}
r_n^{(t)} + \lambda^{(t)} \mathrm{Sgn}\left(x_n^{(t)}\right) = 0 & \qquad \forall n : x_n^{(t)} \neq 0 \\
\left| r_n^{(t)} \right| \leq \lambda^{(t)} & \qquad \forall n : x_n^{(t)} = 0
\end{array}
}
$$

How does it look like? $x_n = 0 \ \forall n$, *but* $n = n_1, n_2, \ldots, n_s$



The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

> *We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ and update $\boldsymbol{x}^{(1)}$ as*
>
> $$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

$\boldsymbol{d}^{(1)}$ *is a 1-sparse vector whose non-zero element is at n_1*

$$d_{n_1}^{(1)} = -\frac{\operatorname{Sgn}\left(r_{n_1}^{(0)}\right)}{\boldsymbol{a}_{n_1}^{\mathsf{T}} \boldsymbol{a}_{n_1}}$$

with $\boldsymbol{a}_{n_1}$ being the n_1-th column of $\mathbf{A}$

The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ *and update* $\boldsymbol{x}^{(1)}$ *as*

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

One could check that for any $0 < \beta^{(1)} \leq \lambda^{(0)}$ *we have*

$$r_{n_1}^{(1)} + \lambda^{(1)} \mathrm{Sgn}\left(x_{n_1}^{(1)}\right) = 0$$

So, $x_{n_1}^{(1)} \neq 0$

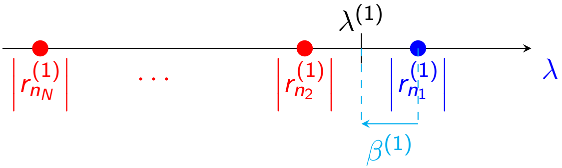
The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ *and update* $\boldsymbol{x}^{(1)}$ *as*

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

But we want to keep the others elements zero! So, we bound $\beta^{(1)}$



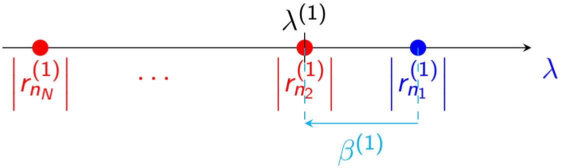
The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ and update $\boldsymbol{x}^{(1)}$ as

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

We try to find the largest possible choice



The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ and update $\boldsymbol{x}^{(1)}$ as

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

One can verify that the largest choice of $\beta^{(1)}$ is

$$\beta^{(1)} = \min \bigcup_{n \neq n_1} Sub^{+} \left\{ \frac{\lambda^{(0)} + r_n^{(0)}}{1 - \left[\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{d}^{(1)}\right]_n}, \frac{\lambda^{(0)} - r_n^{(0)}}{1 + \left[\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{d}^{(1)}\right]_n} \right\}$$

with $Sub^{+}\{\cdot\}$ calculating the subset positive elements

The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ *and update* $\boldsymbol{x}^{(1)}$ *as*

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

The residual is then updated as

$$\boldsymbol{r}^{(1)} = \mathbf{A}^{\mathsf{T}} \left(\mathbf{A} \boldsymbol{x}^{(1)} - \boldsymbol{y} \right)$$

The Homotopy Method: *Derivation*

We now want to move in a direction which leads to $\lambda \to 0$

We set $\lambda^{(1)} = \lambda^{(0)} - \beta^{(1)}$ *and update* $\boldsymbol{x}^{(1)}$ *as*

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \beta^{(1)} \boldsymbol{d}^{(1)}$$

We now update the support $\mathcal{S}^{(2)} = \{n_1, n_2\}$ *with*

$$n_2 = \operatorname{argmin} \bigcup_{n \neq n_1} Sub^+ \left\{ \frac{\lambda^{(0)} + r_n^{(0)}}{1 - \left[\mathbf{A}^\mathsf{T} \mathbf{A} \boldsymbol{d}^{(1)}\right]_n}, \frac{\lambda^{(0)} - r_n^{(0)}}{1 + \left[\mathbf{A}^\mathsf{T} \mathbf{A} \boldsymbol{d}^{(1)}\right]_n} \right\}$$

and exclude it in step 2

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

> *We set* $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ *and update* $\boldsymbol{x}^{(t)}$ *as*
>
> $$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

We have a $\left|\mathcal{S}^{(t)}\right|$-sparse path $\boldsymbol{d}^{(t)}$ with non-zero elements

$$\boldsymbol{z} = -\left(\mathbf{A}_{\mathcal{S}^{(t)}}^{\mathsf{T}} \mathbf{A}_{\mathcal{S}^{(t)}}\right)^{-1} \mathrm{Sgn}\left(\boldsymbol{r}_{\mathcal{S}^{(t)}}^{(t)}\right)$$

This means that we set $\boldsymbol{d}_{\mathcal{S}^{(t)}}^{(t)} = \boldsymbol{z}$ and the other entries of $\boldsymbol{d}^{(t)}$ zero

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

We now find the largest choice of $\beta^{(t)}$ which is

$$\beta_{+}^{(t)} = \min \bigcup_{n \notin \mathcal{S}^{(t)}} Sub^{+} \left\{ \frac{\lambda^{(t-1)} + r_n^{(t-1)}}{1 - \left[\mathbf{A}^{\mathsf{T}} \mathbf{A} \boldsymbol{d}^{(t)}\right]_n}, \frac{\lambda^{(t-1)} - r_n^{(t-1)}}{1 + \left[\mathbf{A}^{\mathsf{T}} \mathbf{A} \boldsymbol{d}^{(t)}\right]_n} \right\}$$

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

This means that the next non-zero entry holds at

$$n_{t+1}^{+} = \operatorname{argmin} \bigcup_{n \notin \mathcal{S}^{(t)}} Sub^{+} \left\{ \frac{\lambda^{(t-1)} + r_n^{(t-1)}}{1 - \left[\mathbf{A}^\top \mathbf{A} \boldsymbol{d}^{(t)}\right]_n}, \frac{\lambda^{(t-1)} - r_n^{(t-1)}}{1 + \left[\mathbf{A}^\top \mathbf{A} \boldsymbol{d}^{(t)}\right]_n} \right\}$$

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

But remember that the residual gets updated in each step

It might be also the case that we get rid of one non-zero entry

So, we should also check this!

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

This happen if step-size

$$\beta_{-}^{(t)} = \min \bigcup_{n \in \mathcal{S}^{(t)} : d_n^{(t)} \neq 0} Sub^{+} \left\{ -\frac{x_n^{(t-1)}}{d_n^{(t)}} \right\}$$

exists and is smaller than $\beta_{+}^{(t)}$

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

> *We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as*
>
> $$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

Could it be the case that $\beta_{-}^{(t)}$ does not exist?

> *Yes! If*
>
> $$\forall n \in \mathcal{S}^{(t)} : d_n^{(t)} \neq 0 : -\frac{x_n^{(t-1)}}{d_n^{(t)}} < 0$$

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

If this happens, we should get rid of the entry at

$$n_{t+1}^{-} = \operatorname{argmin} \bigcup_{n \in \mathcal{S}^{(t)} : d_n^{(t)} \neq 0} Sub^{+} \left\{ -\frac{x_n^{(t-1)}}{d_n^{(t)}} \right\}$$

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

How should we choose $\beta^{(t)}$?

If $\beta_-^{(t)}$ exists and $\beta_-^{(t)} < \beta_+^{(t)}$; then,

- *We set $\beta^{(t)} = \beta_-^{(t)}$*
- *We update $\mathcal{S}^{(t+1)} = \mathcal{S}^{(t)} - \left\{ n_{t+1}^- \right\}$*

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ and update $\boldsymbol{x}^{(t)}$ as

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

How should we choose $\beta^{(t)}$?

Otherwise ...

- *We set $\beta^{(t)} = \beta_+^{(t)}$*
- *We update $\mathcal{S}^{(t+1)} = \mathcal{S}^{(t)} \cup \left\{ n_{t+1}^+ \right\}$*

The Homotopy Method: *Derivation*

We keep repeating this strategy: We are at iteration $t \geq 1$

We set $\lambda^{(t)} = \lambda^{(t-1)} - \beta^{(t)}$ *and update* $\boldsymbol{x}^{(t)}$ *as*

$$\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)} + \beta^{(t)} \boldsymbol{d}^{(t)}$$

We finally update the residual as

$$\boldsymbol{r}^{(t)} = \mathbf{A}^{\mathsf{T}} \left(\mathbf{A} \boldsymbol{x}^{(t)} - \boldsymbol{y} \right)$$

and use it together with $\mathcal{S}^{(t+1)}$ *in the next step*

The Homotopy Method: *Derivation*

We keep repeating this strategy: We stop at iteration s

At iteration s, we should see

$$\lambda^{(s)} = 0$$

Attention!

There might be the case that n_{t+1}^{+} and n_{t+1}^{-} are not unique!

- *In this case, some extra searches should be performed*
- *If interested, check the textbook: Page 480, Remark 15.3*

The Homotopy Method: *Derivation*

We keep repeating this strategy: We stop at iteration s

At iteration s, we should see

$$\lambda^{(s)} = 0$$

Side Note

Why we call it the homotopy method? Because

$$\boldsymbol{d}^{(1)}, \dots, \boldsymbol{d}^{(s)} \text{ describes a homotopy path from } \mathbf{0} \text{ to } \boldsymbol{x}^\star$$

where $\boldsymbol{x}^\star$ is the basis pursuit solution

Final Points

Summary

- ℓ_1-norm minimization gives

  Best relaxation of the optimal recovery

- ℓ_1-norm minimization recovers a sparse signal whose

  $\#$ of non-zero entries $\leq$ $\#$ of samples

- ℓ_1-norm minimization has various forms
  - Basis pursuit
  - Basis pursuit with quadratic constraint
  - Basis pursuit denoising
  - LASSO
- We know how to implement it via the homotopy method

What We Learn Next?

For some applications,

ℓ_1-norm minimization is still computationally complex

what we can do in such applications?

We could go for much lighter approaches

- *Greedy algorithms*
- *Thresholding algorithms*

We are going to learn them in the next part

Which Parts of Textbooks?

We are over with this part

I would suggest to go over the textbook

A Mathematical Introduction to Compressive Sensing
S. Foucart and H. Rauhut, Book, 2013

and study the following parts:

- *Chapter 3: Sections 3.1 and Chapter 15*